View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Assam Don Bosco University Journals

An Energy Efficient Android Application

V. Hurbungs[1], Y. Beeharry[1], A. K. Calkee[1], G. Ahotar[1]

[1]Department of Computer Science and Engineering, University of Mauritius,
Réduit. MAURITIUS.
v.hurbungs@uom.ac.mu; y.beeharry@uom.ac.mu;
avish.calkee@umail.uom.ac.mu, goshik.ahotar@umail.uom.ac.mu

Abstract: ***Statistics demonstrate that Android is the mostly used operating system on mobile phones and tablets across the world. These mobile devices operate using batteries which have limited size and capacity. Therefore, energy management when running mobile applications become of vital importance. In this paper, an energy-efficient android application for goods delivery management system has been developed using Mobile Android Energy Saving Algorithm (MAESA) which is developed based on the best practices presented in this work. The major aim of this application was to monitor and minimise the energy consumption while running the application. Two applications have been built: a normal prototype and an energy-optimised prototype. Seven best practices have been identified and analysis has been performed on energy consumption by both applications developed for four different scenarios over a period of 60 minutes. Results demonstrate that the energy-optimised prototype consumes less energy. For every hour, 597.6 J of energy and 8% of the battery level can be saved with the proper application development and energy optimisation techniques.***

Keywords: Energy consumption, Mobile phones, Android application, Best practices.

1. Introduction

For the past few decades, the world has witnessed a phenomenal evolution of mobile phones and networks, with an estimated cellular subscription of more than 4 billion worldwide in 2009 [1]. Mobile phones started as simple devices for improving communication (phone calls and text messages) between people and have now evolved to a more popular device (called smartphone) possessing multiple functions. Among all the existing operating systems for mobile devices, android is the most used in the world. As such, there are more users worldwide using android operating mobile phones and tablets [2]. Android is an open source mobile operating system and according to the report in [3], the Android market is growing at a very fast rate. With the increasing number of applications being developed on the Android platform and the limited size and capacity of the batteries of mobile devices, significant attention needs to be given to analysing and improving energy consumption of mobile phones when using resource-intensive applications. The energy consumption problem is also extended to several fields in information and communications. For example, in [4], an energy efficient, optimal virtual machine placement (E2OVMP) algorithm has been proposed which minimizes the expenses for hosting virtual machines in a cloud provider environment.

A. Literature Review

Global Positioning System (GPS) is a promising solution for location-based services such as fleet management, tracking of vehicles and location finder. However, GPS services were available to manufacturers only due to proprietary issues in the past. With the introduction of android-based open source operating system for mobile devices, the users have now the possibility to access the mobile hardware directly [5]. As per the report published by the Global Industry Analysts Inc. [6], GPS Market Worldwide is projected to reach $28.9 billion by 2015. Along this line, several applications which use GPS have been developed. In [7], the android application from TomTom (leading company in navigation systems) has been analysed using a forensic acquisition mechanism. In order to tackle the problem of passengers missing their bus in the city of Nasik, the authors of [8] have proposed an android application for the public transport system to provide dynamic information to the user about current locations and timings of the buses using GPS. In [9], an energy-efficient proximity alert service for android has been designed and presented. The application presented reduces the GPS usage by 96.66% and increases the lifetime of the battery by 75.71% as opposed to baseline proximity alert in android. In view to solve the queueing issue when buying suburban railway tickets, Android Suburban Railway (ASR) ticketing has been presented in [10] where the user can make a railway ticket reservation with his smart phone and carry his railway tickets as a QR-code. GPS is employed to automatically authenticate and erase the QR-code based ticket after a specific time interval upon reaching the final destination. Finally, in [11] an automated android ticketing system application has been presented which automatically identifies the passenger and deducts the fare depending on the distance travelled.

B. Energy Consumption of Smart Phones and Applications

In [11], a survey has been carried out on the energy consuming components of a mobile device such as wireless interfaces (Bluetooth, Wi-Fi, 2G / 3G / 4G), display, music / video players. The results and measurements presented in this survey give a thorough understanding of the intensity of energy consumption by the different parts of a mobile device. Also, the main findings revealed that the wireless technologies are the most energy consuming parts. In [12], the dynamics of the interdependencies between resources have been analysed and the results demonstrate that making use of simple scheduling algorithms do not lead to optimum usage of those resources. Some trends are being followed in order to maximise the energy storing capacities of batteries for mobile devices [11]. For instance, in [13], researchers are making use of nano-technology to make batteries which are able to produce more electricity than existing lithium-ion batteries. Other researchers are trying to exploit the movement of the user for recharging the battery of the phone [14]. The authors in [1], have developed TailEnder which is a protocol with the ability to minimise the energy consumption of mobile applications while not having a great impact on meeting user-specified deadlines.

Given the remarkable importance given to GPS based applications and the optimisation of energy consumption of smartphones, an energy-efficient android application for goods delivery management system in Mauritius has been developed in this work. Seven best-practices have been incorporated in the android application so as to improve the energy efficiency and reduce the battery drainage of the smartphone. The best-practices are: using the finish method to kill unused activities; using the screen on method; reducing the brightness and MAP display when a pre-defined battery threshold has been reached; using Fused Location Provider; using a black wallpaper; using good coding practices and connecting to Wi-Fi only when the application requires it. The application was built using Java programming language on Eclipse development environment.

C. Overview of Experiments Performed

Experiments were performed to determine the energy consumption of both the conventional and energy-optimised application for four different scenarios whereby different applications, namely, Viber, Radio and Internet Browser are run in parallel. Two phone models are used for testing purposes. The first one is an HTC Desire 310 with Android OS version 4.2.2, quad-core processor 1.3 GHz Cortex-A7, Display size 480 x 854 pixels, 2.0 Gb Memory and with A-GPS. The second phone is an LG L90 D405 with Android OS version 4.4.2, quad-core processor 1.2 GHz Cortex-A7, Display size 540 x 960 pixels, 3.87 Gb Memory and with GPS. The application used for real-time power consumption measurement is PowerTutor2 [12]. Using the data obtained from the tests performed, it has been found that the energy-optimised application can save 597.6 J of energy and 8% of the battery level every hour with the proper application development and energy optimisation techniques.

The organisation of this paper is as follows. Section 2 describes the complete system model of the goods delivery management system. The android application is described and the best-practices deployed in the building of the energy-optimising application are explained. Section 3 presents the results obtained with discussions and Section 4 concludes the paper.

2. System Model

This section describes the complete system model of the android application for goods delivery management system in Mauritius. The proposed system architecture is described and an overview of the software used for the system development is given. The seven best-practices used in the development of the energy-optimised application are also described.

2.1 System Architecture

Figure 1 shows the complete system architecture. In this model, the native applications interact directly with the device's operating system and are coded in Android Java. The communication is thus done directly with the GPS of the device in order to obtain the location of the vehicle. The client side consists of an android application and a website. The application is built for android operated mobile phones using Java programming language on the Eclipse development environment and Google MAP API v1. The Server side is built using PHP as open source scripting language. The Database used at the client side for local storage is SQLite since it is integrated in Android mobile operating systems. At the server side, MySQL database is used in order to ease the deployment of the system.

Figure1: Complete System Architecture.

Figure 2 shows the android application interface and login process for the vehicle's driver.

Figure2: Android Application Interface for the Driver.

The GPS activity and route to be covered by the driver can be viewed by selecting the GPS menu as shown in Figure 3.

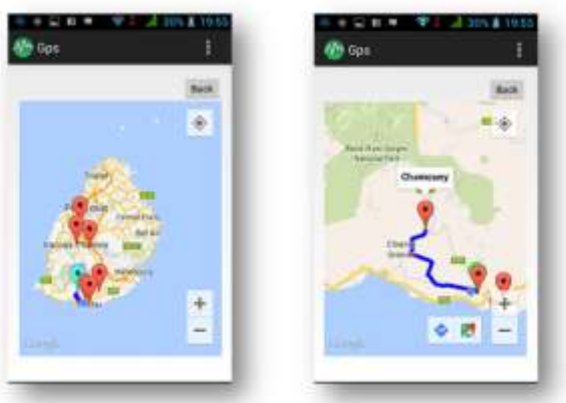

Figure3: GPS Activity and Route to be covered by the Driver

2.2 Energy Consumption Optimization Best Practices

In the optimised version of the android application, seven best practices described in Section 1 above have been deployed. These best practices are described in the next sub-sections. The proposed algorithm based on these best practices is named: Mobile Android Energy Saving Algorithm (MAESA).

1) The Finish Method

The Finish method closes previously used activities that the user created when using the application which thus helps in energy consumption reduction. Using the Finish method ensures only required processes are active within the application.

2) Screen On

The Screen On state allows the screen to remain turned on while the user is using the application on the smartphone. The screen turned off when it is not being used.

3) Brightness and Map Display Reduction

The system automatically lowers the brightness level of the smartphone and reduces the map details displayed when the battery level is below or equal to 50%. The threshold is set to 50% in this work due to the visibility issues with the HTC mobile phone. However, the threshold can be set to a higher value in order to reduce even more the energy consumption.

4) Fused Location Provider

The Fused Location Provider provides the best location to the users according to their needs in an energy-efficient way [13].

5) Black Wallpaper

A black wallpaper saves energy in two different ways [17]:

1. The best screens do not need energy when using black pixels and
2. The black wallpaper does not load any image, which thus aids the optimal usage of the CPU and therefore improving the smartphone's performance and reducing the energy consumption.

6) Good Coding Practice

As has been demonstrated in [14], good coding practice is of utmost importance for the building of energy-efficient applications. It has been found in [14] that grouping network packets and using good coding practices for reading arrays, accessing variables and performing invocations could result in the reduction of energy consumption by smartphones.

7) Wi-Fi On and Off

The WIFI_MODE_FULL method is used to automatically build a connection to a range of access points which are remembered. Also, using the proper methods, the Wi-Fi connection is turned off automatically when the application is not being used.

2.3 Implementation of MAESA with the Android Application

Figure 4 shows the system architecture with the best practices implemented with the mobile application using the MAESA algorithm.

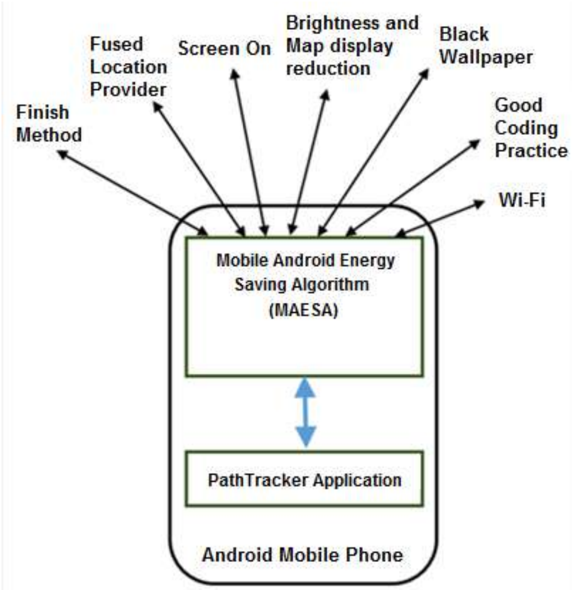

Figure4: the system architecture with the best practices implemented with the mobile application using the MAESA algorithm

The seven best practices described have been implemented in the application to allow for energy saving. The method for brightness adjustment and Map details checks the battery level of the android mobile device. When the battery level goes below 50%, the screen's brightness is automatically reduced. Additionally, the Map details displayed are reduced. The finish method kills any previous unused activities when using the application. The method which handles the Wi-Fi simply checks the state of the signal. If data is being transmitted or received, the Wi-Fi stays active, otherwise it is turned off automatically.

3. Results

The evaluation of the system is performed by comparing and contrasting between the conventional and energy-optimised prototypes. This comparison is based on the energy consumption of both applications over a period of 60 minutes, split into 4 time intervals of 15 minutes. 4 scenarios have been devised and are explained below [19]:

- **Scenario 1**: Running the PathTracker application only.
- **Scenario 2**: Running the PathTracker application and Viber.
- **Scenario 3**: Running the PathTracker application, Viber and Radio.
- **Scenario 4**: Running the PathTracker application, Viber, Radio and Internet Browser.

Scenario 1 is run between 0 to 15 minutes. Scenario 2 is run between 15 minutes to 30 minutes. Scenario 3 is run between 30 minutes to 45 minutes and finally Scenario 4 is run from 45 minutes to 60 minutes. The measurements are performed in the following order:

1. The energy consumption and battery usage of the conventional application for the 4 scenarios over 1 hour are measured, plotted and analysed.
2. The same measurements are performed for the conventional application with each of the 7 best-practices incorporated separately.
3. The measurements and analysis for the energy-optimised application which is a combination of all the 7 best-practices are performed.

Figure 5 shows the energy consumption over 1 hour for the 4 different scenarios when the conventional application is used. It can be observed that the energy consumed with the conventional application rises from 5.4 Joules to 1600 Joules over 1 hour. Figure 6 shows the battery usage profile over 1 hour for the 4 scenarios. Clearly, it can be observed that the battery level decreases from 69% to 47% over 1 hour. The decrease in battery life with the conventional application for every hour is 22%. The high rate of energy consumption during Scenario 1 is due to the energy hungry GPS application. The decrease in the rate of energy consumption when switching from Scenario 1 to Scenario 2 is due to the fact that the conventional prototype starts to run in the background and goes to a temporary sleep mode. This decrease in energy consumption rate is observed in all the tests performed.

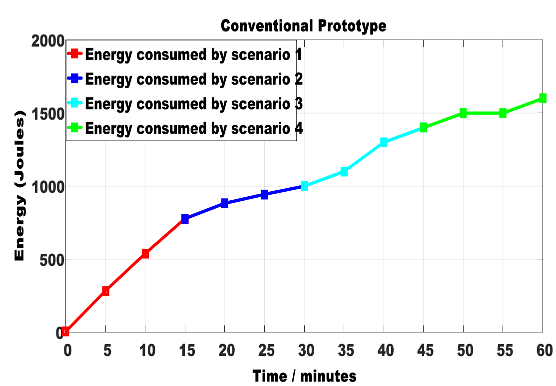

Figure5: Energy consumption with the conventional application

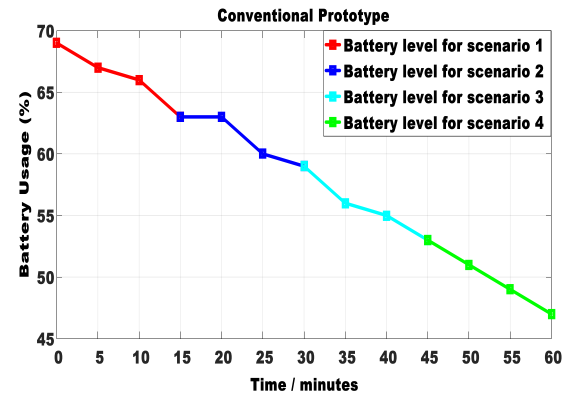

Figure6: Battery usage profile with the conventional application

Figure 7 shows the energy consumption over 1 hour for the 4 different scenarios when the conventional application is used with the finish method best-practice only. It can be observed that the energy consumed with the conventional application rises from 2.7 Joules to 1120 Joules over 1 hour. Figure 8 shows the battery usage profile over 1 hour for the 4 scenarios. It can be observed that the battery level decreases from 47 % to 22 % over 1 hour. The decrease in battery life for the application optimised with only the finish method best-practice is 25% for every hour. The application used to measure the energy consumption of the running applications on the mobile device can only give measurements of energy consumption for each application running separately. Due to this limitation, higher values of battery drainage are observed when the finish method and the screen on best-practices are used. The values for the battery drainage consist of other inner operations like updates running in parallel when the tests were performed.

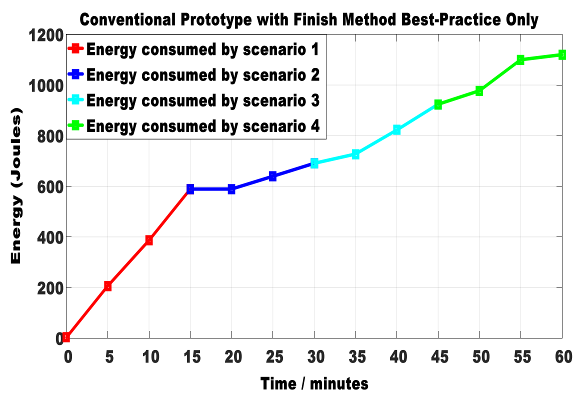

Figure7: Energy Consumption with Finish Method Best-Practice Only

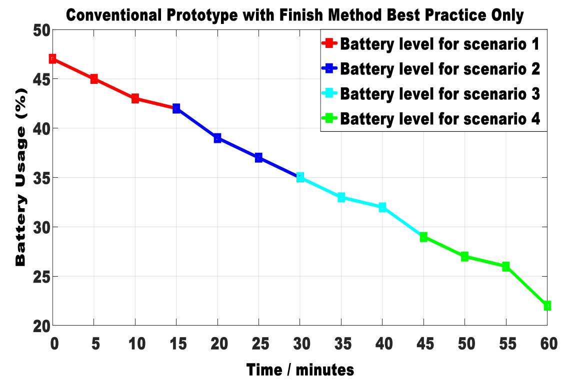

Figure8: Battery Usage Profile with Finish Method Best-Practice Only

Figure 9 shows the energy consumption over 1 hour for the 4 different scenarios when the conventional application is used with the screen on best-practice only. It can be observed that the energy consumed with the conventional application rises from 2.7 Joules to 1100 Joules over 1 hour. Figure 10 shows the battery usage profile over 1 hour for the 4 scenarios. It can be observed that the battery level decreases from 73 % to 50 % over 1 hour. The decrease in battery life for the application optimised with only the finish method best-practice is 23% for every hour.

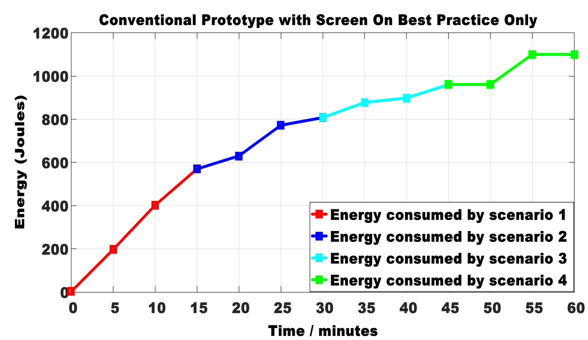

Figure9: Energy Consumption with Screen On Best-Practice Only

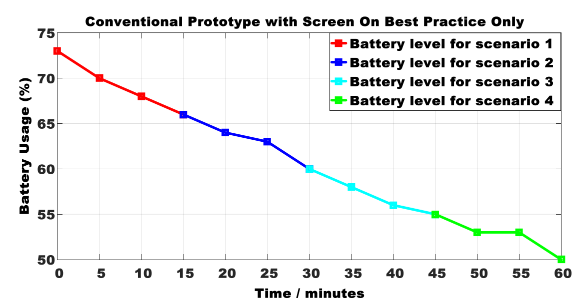

Figure10: Battery Usage Profile with Screen On Best-Practice Only

Figure 11 shows the energy consumption over 1 hour for the 4 different scenarios when the conventional application is used with the brightness reduction best-practice only. It can be observed that the energy consumed with the conventional application rises from 1.6 Joules to 701.5 Joules over 1 hour. Figure 12 shows the battery usage profile over 1 hour for the 4 scenarios. It can be observed that the battery level decreases from 41 % to 22 % over 1 hour.

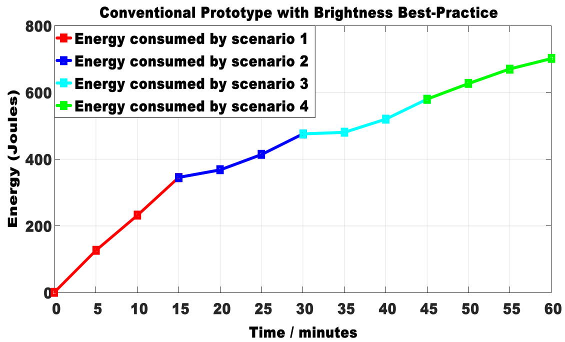

Figure11: Energy Consumption with Brightness Reduction Best-Practice Only

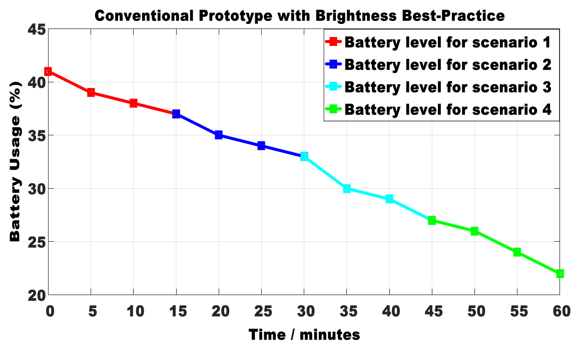

Figure12: Battery Usage Profile with Brightness Reduction Best-Practice Only

Figure 13 shows the energy consumption over 1 hour for the 4 different scenarios when the conventional application is used with the black wallpaper best-practice only. It can be observed that the energy consumed with the conventional application rises from 0.908 Joules to 1400 Joules over 1 hour. Figure 14 shows the battery usage profile over 1 hour for the 4 scenarios. It can be observed that the battery level decreases from 90 % to 70 % over 1 hour.

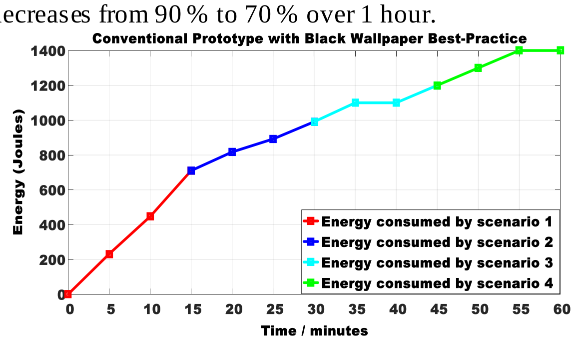

Figure13: Energy Consumption with Black Wallpaper Best-Practice Only

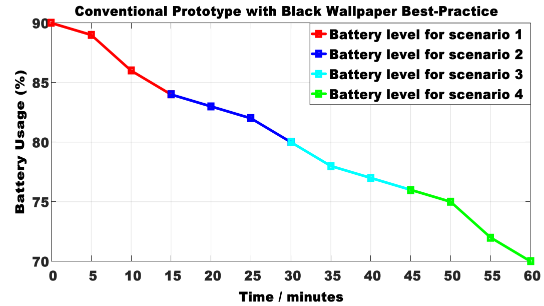

Figure14: Battery Usage Profile with Black Wallpaper Best-Practice Only

Figure 15 shows the energy consumption over 1 hour for the 4 different scenarios when the conventional application is used with the Fused Location Provider best-practice only. It can be observed that the energy consumed with the conventional application rises from 1.8 Joules to 1500 Joules over 1 hour. Figure 16 shows the battery usage profile over 1 hour for the 4 scenarios. It can be observed that the battery level decreases from 98 % to 78 % over 1 hour.

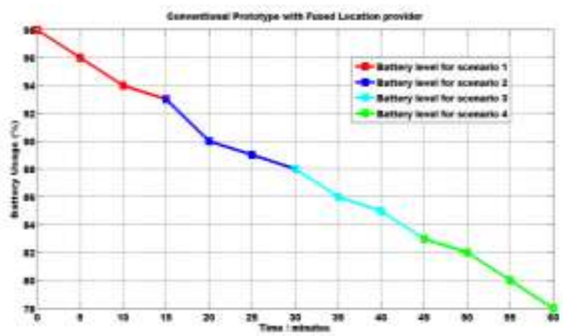

Figure15: Energy Consumption with Fused Location Provider Best-Practice Only

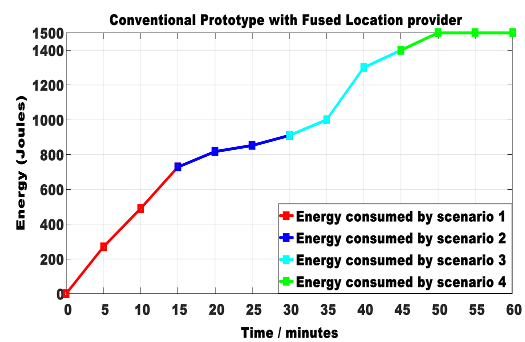

Figure16: Battery Usage Profile with Fused Location Best-Practice Only

Figure 17 shows the energy consumption over 1 hour for the 4 different scenarios when the conventional application is used with the good coding best-practice only. It can be observed that the energy consumed with the conventional application rises from 1.8 Joules to 1200 Joules over 1 hour. Figure 18 shows the battery usage profile over 1 hour for the 4 scenarios. It can be observed that the battery level decreases from 76 % to 57 % over 1 hour.

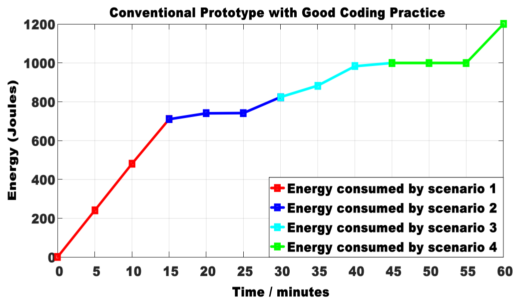

Figure17: Energy Consumption with Good Coding Best-Practice Only

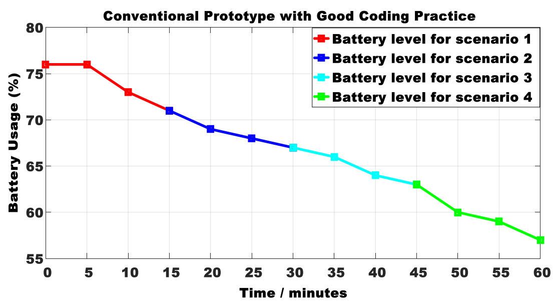

Figure18: Battery Usage Profile with Good Coding Best-Practice Only

Figure 19 shows the energy consumption over 1 hour for the 4 different scenarios when the conventional application is used with the Wi-Fi on / Wi-Fi off best-practice only. It can be observed that the energy consumed with the conventional application rises from 4 Joules to 1500 Joules over 1 hour. Figure 20 shows the battery usage profile over 1 hour for the 4 scenarios. It can be observed that the battery level decreases from 95 % to 75 % over 1 hour.

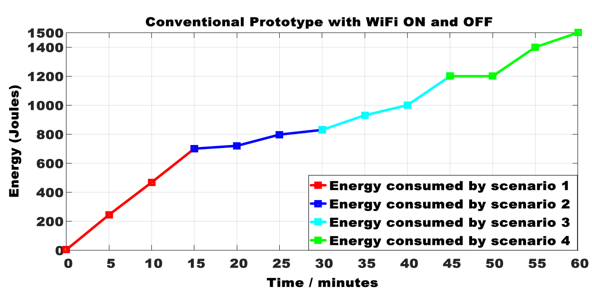

Figure19: Energy Consumption with Wi-Fi On and Wi-Fi Off best-Practice Only

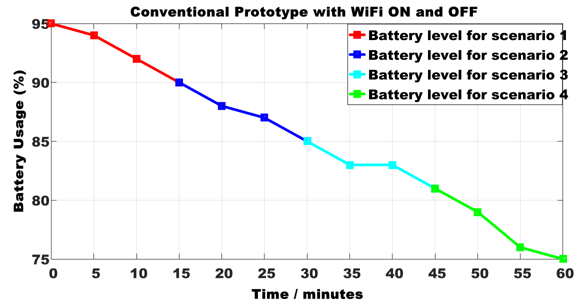

Figure20: Battery Usage Profile with Wi-Fi On and Wi-Fi Off best-Practice Only

Figure 21 shows the energy consumption over 1 hour for the 4 different scenarios when the energy-optimised application (with all the 7 best-practices incorporated) is used. It can be observed that the energy consumed with the conventional application rises from 3 Joules to 1000 Joules over 1 hour. Figure 22 shows the battery usage profile over 1 hour for the 4 scenarios. It can be observed that the battery level decreases from 53 % to 39 % over 1 hour.

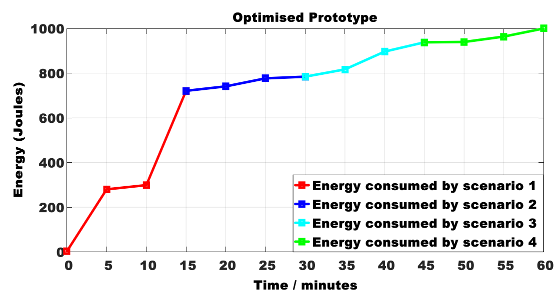

Figure21: Energy Consumption with Energy Optimised Prototype

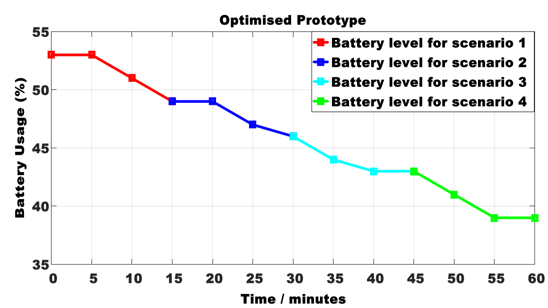

Figure22: Battery Usage Profile with Energy Optimised Prototype

A comparison between the energy consumed by the conventional and energy-optimised prototypes during 1 hour with the 4 scenarios is shown in Table 1. The tests have been performed while making sure that only the prototypes and the different applications for each scenarios are run so as to obtain the most precise impact on energy consumption values and the battery drainage levels.

TABLE 1: COMPARISON BETWEEN CONVENTIONAL AND ENERGY-OPTIMISED PROTOTYPES

	Conventional Prototype	**Optimised Prototype**
Energy / J (at time = 0 minutes)	5.4	3
Energy / J (at time = 0 minutes)	1600	1000
Energy Consumed / J	1594.6	997
Battery Level / % (at time = 0 minutes)	69	53
Battery Level / % (at time = 60 minutes)	47	39
Battery Level Difference / %	22	14

It can be clearly observed from Table 1 that he conventional prototype consumes much higher energy than the energy-optimised prototype. This higher energy consumption has a direct impact on the battery drainage as well. The energy-optimised prototype consumes 597.6 J less and helps save 8% of battery level every hour.

4. Conclusion

The main aim of this work was to develop an energy-optimised application for a GPS based goods delivery management system in Mauritius. First, a conventional android application together with the system administrator's webpage were designed and implemented. Then, 7 best-practices which are useful in android application energy-optimisation were deployed for the implementation of the optimised application. Both applications were tested for energy consumption and battery drainage over 1 hour with 4 different scenarios at steps of 15 minutes. These scenarios involve the running of Viber, Radio and Internet Browser applications in parallel with the applications. The results demonstrate that the energy-optimised application can aid to an energy consumption reduction of about 597.6 Joules and also 8% of battery life can be saved every hour.

The major achievement in this work is the development of an energy-optimised android application for goods delivery management system in Mauritius. Such a system shall help goods delivery companies to properly keep track of the goods delivered and vehicles in the most energy-efficient manner. With the current trend of the increasing use of android mobile applications and problem with high energy consumption, it is imperative to find proper solutions for developing applications which can help significantly in reducing the fast drainage of battery levels. The assessment provided in this work could create awareness in the android coding community so that more energy-efficient GPS-based applications are developed and also, incite researchers for finding new techniques which will reduce energy consumption when implemented in android applications.

5. Acknowledgement

The authors would like to thank the University of Mauritius for all the facilities provided.

References

[1] N. Balasubramanian, A. Balasubramanian and A. Venkataramani, "Energy consumption in mobile phones: a measurement study and implications for network applications," in *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference*, New York, USA, 2009.

[2] G. Pandey and D. Dani, "Android Mobile Application Build on Eclipse," *International Journal of Scientific Research Publications* , vol. 4, no. 2, pp. 1 - 5, 2014.

[3] E. Mills, "Report: Android app market growing faster than iPhone apps," 16 February 2011. [Online]. Available: http://www.cnet.com/news/report-android-app-market-growing-faster-than-iphone-apps/. [Accessed 05 April 2015].

[4] E. Barlaskar, N. A. Singh and Y. J. Singh, "Energy optimisation methids for Virtual Machine Placement in Cloud Data Center," *ADBU Journal of Engineering and techology*, pp. 1 - 7, 2014.

[5] S. Kumar, M. A. Qadeer and A. Gupta, "Location based services using android (LBSOID)," in *IEEE International Conference on Internet Multimedia Services Architecture and Applications (IMSAA)*, Bangalore, 2009.

[6] Global Industry Analysts Inc., "PRWeb," Global Industry Analysts Inc., 20 October 2010.

[Online]. Available: http://www.prweb.com/releases/global_positioning_system/GPS/prweb4670914.htm. [Accessed 7 April 2015].

[7] M. Roeloffs, N. Le-Khac and T. Kechadi, "Forensic Investigation of Tomtom Applcation on Android Mobile Devices," in *10th Annual IFIP WG International Conference on Digital Forensics*, Austria, 2014.

[8] S. Agrawal, N. Ahire and S. Sonavane, "GPS Supported Android Application for City Bus Scheduling and Tracking System," *International Journal of Enhanced Research in Management and Computer Applications*, vol. 3, no. 12, pp. 26 - 29, 2014.

[9] M. F. Bulut and M. Demirbas, "Energy efficient proximity alert on Android," in *IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, San Diego, CA, 2013.

[10] S. Khoja and M. Kadam, "Android sub-urban railway ticketing using GPS as ticket checker," *International Journal of Technical Research and Applications*, vol. 2, no. 3, pp. 169 - 174, 2014.

[11] T. Dongare, A. Babar and M. Nivangune, "Android Application for Ticket Reservation with GPS as Ticket Valication," *International Journal of Engineering Research in Management and Technology*, vol. 3, no. 3, pp. 138 - 141, 2014.

[12] PowerTutor, "A Power Monitor for Android-Based Mobile Platforms," PowerTutor, 5 November 2014. [Online]. Available: http://ziyang.eecs.umich.edu/projects/powertutor/instructions.html. [Accessed 09 April 2015].

[13] Google, "Android Developers," Google, [Online]. Available: https://developer.android.com/google/play-services/location.html. [Accessed 12 April 2015].

[14] D. Li and W. G. J. Halfond, "An Investigation into Energy-Saving Programming Practices for Android Smartphone App Development," in *Proceedings of the 3rd International Workshop on Green and Sustainable Software (GREENS 2014)*, New York, 2014.

[15] G. P. Perrucci, F. H. P. Fitzek and J. Widmer, "Survey on Energy Consumption Entities on the Smartphone Platform," in *Proceedings of the 73rd IEEE Vehicular Technology Conference (VTC Spring)*, Yokohama, Japan, 15 - 18 May 2011.

[16] N. Vallina-Rodriguez, P. Hui, J. Crowcroft and A. Rice, "Exhausting Battery Statistics-Understanding the energy demands on mobile handsets," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Networking, Systems, and Aplications on Mobile Handhelds*, New York, 2010.

[17] "Nanowire battery can hold 10 times the charge of existing lithium-ion battery," Stanford Technical Report, Stanford, 2007.

[18] Q. Naing, V. Hoffer, J. A. Weber, D. J. Kuo, A. D. Donelan and J. M. Li, "Biomedical energy harvesting: Generating electricity during walking with minimal user effort," *Science*, vol. 319, no. 5864:807, 2008.

[19] Google, "Google play," Google, 2015. [Online]. Available: https://play.google.com/store/apps/details?id=net.rocboronat.android.wallpaper.npe&hl=en. [Accessed 09 April 2015].

[20] A. Calkee and G. Ahotar, *Reducing Power Consumption of Android Application (Unpublished Bachelor's Thesis)*, Réduit, Mauritius: University of Mauritius, 2014.

Author Profile

Visham Hurbungs, BSc. Information Systems, MSc. Computer Science with specialisation in Software Engineering at the University of Mauritius. Joined the University of Mauritius as research assistant in 2004 and has subsequently occupied the post of Programmer/Analyst and Systems Analyst. He joined as lecturer in the Computer Science and Engineering department in 2010. Prior to joining the academic, he has acquired professional experience in software/web development, website administration and maintenance. He has also been involved in the installation/ configuration of e-learning platforms both in a local and international context. As an academic, his main teaching areas and research interests are: Web programming, Object-Oriented Software Development, Database development and Mobile applications. Visham Hurbungs has been a member of the technical programme committee of the IEEE Africon 2013 conference held in Mauritius. He also acted as reviewer and co-chair of the Tutorial/Workshop sub-committee of Africon 2013. He is currently a member of the technical programme committee and the chair of the Submissions

Committee for ELECOM 2016 (Springer LNEE Series).

Yogesh Beeharry, BEng Electronics and Communications Engineering, is currently a PhD student in the field of error control coding at the University of Mauritius. He holds a BEng (Hons) Electronics and communications Engineering with First class from the University of Mauritius. His main research interests are: error control coding, Turbo codes, mobile communications and source and channel coding.

Avish K. Calkee, BSc. Computer Applications, has recently completed his undergraduate degree from the University of Mauritius. He is currently working as Android mobile developer at Esokia Web and Mobile Agency.

Goshik Ahotar, BSc. Computer Applications, has recently completed his undergraduate degree from the University of Mauritius. He is currently working as Software Engineer at Proximity BBDO Indian Ocean.